MISSION OPERATIONS COST REDUCTION BY SOFTWARE INHERITANCE

John A. Rohr
Keyur C. Pate]
Jet Propulsion Laboratory
Pasadena, California USA
John.A.Rohr@Jpl.Nasa.Gov
(818) 354-4096

March 26, 1995

Since the initiation of deep-space exploration by JPL spacecraft in the 1960's, almost all JPL mission operations software has been written for and operated on mainframe computers. in the late 1980's, a phenomenon began which would have a profound effect on JPL mission operations software. At this time, mainframe computers were becoming increasing expensive to operate when compared with workstations such as Sun, Digital Equipment, and other manufacturers. in addition, software written for a mainframe, even if written in a common high-level language such as FORTRAN, was designed to operate with a specific mainframe computer operating system. Converting such software to another mainframe system, when necessary, required a considerable expenditure of resources just to convert to a different operating system, even if the functional program code were unchanged. Most workstations support the UNIX operating system. Therefore, moving from one workstation to another for purposes of upgrading or expansion would require only minimal changes to the software due to minor differences in the implementation of the UNI X operating system on a particular vendor's hardware platform. This open system commonality was appealing to the designers and implementors of JPL mission opera t i ons software.

Furthermore, in the late 1980's JPL began to investigate the concept of building a set of operations software which could be used by multiple missions and would require only a minimal amount of effort for adaptation from one mission to another. The concept of multimission software was appealing to management, which realized that considerable savings could be realized by having a core set of software which could be mostly reused from mission to mission with only minor adaptation required.

The combination of the availability and capability of workstations and the concept of multimission software led to the JPL Multimission Software Transition Project (MSTP). The project officially began in 1993 with an expected completion by the end of 1 996. The project has two objectives. The first is to convert mission operations software from mainframe computers which had been utilized in the past to UNIX workstations. The second is to convert the software in such a way that the resulting software will be multimission in nature and will require only minimum adaptation from mission to mission.

Two specific examples illustrate the MSTP: the Sequence Translator program (SEQTRAN) and the COMSIM data system simulator program. SEQTRAN is a program which is used to translate spacecraft commands in the form of mnemonic instructions with parameters into the binary data required onboard the spacecraft to drive the command and sequencing interpreter. SEQTRAN also includes other functions such as management of onboard memory and specialized output file formatting. Since its inception, SEQTRAN has been programmed and run on UNISYS 1100-family mainframes. Because SEQTRAN was based code inherited from earlier work, it is written entire] y in UNISYS 1100 assembly language. There is a slightly different version of SEQTRAN for each mission. One task of the MSTP is the conversion of SEQTRAN to the C language running under the UNIX operating system on Sun and Hewlett-Packard workstations. The new version of SEQTRAN will also provide one program which can be used for all missions. Configuration data will internally adapt SEQTRAN for each project based in information supplied when the program is invoked.

Several JPL missions have utilized data system simulators to develop flight soft ware, to test sequences before sending them to the spacecraf t, and to investigate anomalies which occur in flight. The simulators model the onboard hardware and simulate the environment so that actual flight software can be run instruction-by-instruction in the simulator. The use of faster onboard processors by recent missions generally means that a simulation of the onboard data system will run much slower than real-time. A research effort was initiated several years ago to investigate the use of fast, multiple-processor workstations to build a data system simulator which would run at least as fast as real time. Following several years of research, a multimission fast simulator framework has been developed which will provide a basis for building data system simulators for several missions, Also, a version of the fast simulator has been built for the Galileo Project. The Galileo version was done outside MSTP funding.

The Voyager Project has utilized the COMSIM data system simulator since before launch to develop flight software and test sequences which are to be sent to the spacecraft. Voyager personnel have also made extensive use of COMSIM for anomaly investigation since the launch of the Voyager spacecraft. COMSIM runs on UNISYS 1100-family mainframes. It is written in FORTRAN and UNISYS 1100 assembly language. COMSIM is one of the programs which has been planned for conversion to UNIX workstations as a part of MSTP. Originally, the FORTRAN code was to be ported directly and the assembly language was to be rewritten in FORTRAN or C.

Because of the success of the Hi-Speed Simulator research effort, the implementation of a Hi-Speed Simulator for Galileo, and a planned implementation of a Hi-Speed Simulator for Cassini, it was decided to drop the effort to convert COMSIM to workstations and build an adaptation of the Hi-Speed Simulator instead. The will not only save development funds, but it also will also give the Voyager project a data system simulator which is much more modern and capable than COMSIM and maintainable in a multimission mode. The decision to use an adaptation of the Hi-Speed Sim rather than converting COMSIM is a win-win situation in which the MSTP reduces costs and the Voyager Project get a simulator with more speed and capability than would be obtained from a conversion of COMSIM.

The use of multimission software will result in software which is cheaper to develop and maintain and which can be transferred to new, upgraded computer systems with minimal effort. The MSTP is the vehicle which will produce some of these software systems for JPL.